```
                                  L 12


   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
SSS              YYY         YYY  SSS
SSS              YYY         YYY  SSS
SSS              YYY         YYY  SSS
SSS                 YYY   YYY     SSS
SSS                 YYY   YYY     SSS
SSS                 YYY   YYY     SSS
   SSSSSSSSS           YYY           SSSSSSSSS
   SSSSSSSSS           YYY           SSSSSSSSS
   SSSSSSSSS           YYY           SSSSSSSSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
```

```
                                                                          N 15
**FILE**ID**RMSVECTOR




RRRRRRRR    MM      MM    SSSSSSSS  VV      VV  EEEEEEEEEE    CCCCCCCC  TTTTTTTTTT    OOOOOO    RRRRRRRR
RRRRRRRR    MM      MM    SSSSSSSS  VV      VV  EEEEEEEEEE    CCCCCCCC  TTTTTTTTTT    OOOOOO    RRRRRRRR
RR      RR  MMMM  MMMM  SS          VV      VV  EE          CC              TT      OO      OO  RR      RR
RR      RR  MMMM  MMMM  SS          VV      VV  EE          CC              TT      OO      OO  RR      RR
RR      RR  MM  MM  MM  SS          VV      VV  EE          CC              TT      OO      OO  RR      RR
RR      RR  MM  MM  MM  SS          VV      VV  EE          CC              TT      OO      OO  RR      RR
RRRRRRRR    MM      MM    SSSSSS    VV      VV  EEEEEEEE    CC              TT      OO      OO  RRRRRRRR
RRRRRRRR    MM      MM    SSSSSS    VV      VV  EEEEEEEE    CC              TT      OO      OO  RRRRRRRR
RR  RR      MM      MM          SS  VV      VV  EE          CC              TT      OO      OO  RR  RR
RR  RR      MM      MM          SS  VV      VV  EE          CC              TT      OO      OO  RR  RR
RR    RR    MM      MM          SS    VV  VV    EE          CC              TT      OO      OO  RR    RR        ....
RR    RR    MM      MM          SS    VV  VV    EE          CC              TT      OO      OO  RR    RR        ....
RR      RR  MM      MM  SSSSSSSS        VV      EEEEEEEEEE    CCCCCCCC      TT        OOOOOO    RR      RR      ....
RR      RR  MM      MM  SSSSSSSS        VV      EEEEEEEEEE    CCCCCCCC      TT        OOOOOO    RR      RR      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
00000001  0000   RMSSWITCH=1                          ;GENERATE RMS SERVICE CASE BRANCH TABLE
          0000          .NLIST  CND
          0000          .TITLE  SYS$RMS_VECTOR - RMS SERVICE VECTOR DEFINITIONS
          0000          .IDENT  'V04-000'
          0000
          0000   ;
          0000   ;****************************************************************************
          0000   ;*                                                                          *
          0000   ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
          0000   ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
          0000   ;*  ALL RIGHTS RESERVED.                                                    *
          0000   ;*                                                                          *
          0000   ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
          0000   ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
          0000   ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
          0000   ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
          0000   ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
          0000   ;*  TRANSFERRED.                                                            *
          0000   ;*                                                                          *
          0000   ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
          0000   ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
          0000   ;*  CORPORATION.                                                            *
          0000   ;*                                                                          *
          0000   ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
          0000   ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
          0000   ;*                                                                          *
          0000   ;*                                                                          *
          0000   ;****************************************************************************
          0000   ;
          0000   ; D. N. CUTLER 22-JUN-76
          0000   ;
          0000   ;  MODIFIED BY:
          0000   ;
          0000   ;       V03-041 LJK0287         Lawrence J. Kenah        2[illegible]-Jun-1984
          0000   ;               Add R5 to entry mask for $CANEXH system service.
          0000   ;
          0000   ;       V03-040 LMP0239         L. Mark Pilant,          23-Apr-1984  9:21
          0000   ;               Change $CHKPRO from an exec mode service to a kernel mode
          0000   ;               service.  This was made necessary by the $CHKPRO (internal
          0000   ;               entry point) interface change.
          0000   ;
          0000   ;       V03-039 MMD0250         Meg Dumont,      27-Feb-1984  17:49
          0000   ;               Add support for $MTACCESS installation specific accessibility
          0000   ;               routine
          0000   ;
          0000   ;       V03-038 DAS0001         David Solomon            20-Feb-1984
          0000   ;               Implement new design for RMS echo SYS$INPUT to SYS$OUTPUT
          0000   ;               (vs V03-019). Echo is now performed by a caller's mode AST
          0000   ;               routine declared in RMS\RMS$EXRMS. Change INCB/DECB of FAB/RAB
          0000   ;               busy bit to BISB/BICB, now that we have room.
          0000   ;
          0000   ;       V03-037 SSA0004         Stan Amway               28-Dec-1983
          0000   ;               For $SETPFM, changed number of parameters from 1 to 4
          0000   ;               and changed entry mask to save R2-R11.
          0000   ;
          0000   ;       V03-036 TMK0002         Todd M. Katz             19-Nov-1983
          0000   ;               The entry point for $ASCTOID can no longer be reached as a
```

```
0000  ;                  branch destination from the executive mode dispatcher.
0000  ;                  A temporary entry point (EXE$ASCTOID) has been placed within
0000  ;                  this module, and a JMP is made from it to the real system
0000  ;                  service entry point (EXE$$ASCTOID).
0000  ;
0000  ;                  Also, change the entry mask for SYS$TRNLOG, so that R8 is
0000  ;                  now saved.
0000  ;
0000  ;          V03-035 TMK0001          Todd M. Katz              22-Oct-1983
0000  ;                  The entry points for $FINISH_RDB and $IDTOASC can no
0000  ;                  longer be reached as branch destinations from the executive
0000  ;                  mode dispatcher. Temporary entry points (EXE$FINISH_RDB and
0000  ;                  EXE$IDTOASC) have been placed within this module, and from
0000  ;                  each a JMP is made to the real system service entry points
0000  ;                  (EXE$$FINISH_RDB and EXE$$IDTOASC).
0000  ;
0000  ;          V03-034 PRB0254          Paul Beck         15-Sep-1983  14:49
0000  ;                  (1) Correct the way synchronous CJF services are defined.
0000  ;                  (2) Define loadable RUF services.
0000  ;
0000  ;          V03-033 WMC0029          Wayne Cardoza             31-Aug-1983
0000  ;                  Loadable services should not be unconditionally inhibited.
0000  ;                  Add an alternate CHMx argument to LDBSRV.
0000  ;
0000  ;          V03-032 DWT0125          David W. Thiel            22-Aug-1983
0000  ;                  Remove CHECKARGLIST and calls to same.
0000  ;
0000  ;          V03-031 MKL0167          Mary Kay Lyons            19-Aug-1983
0000  ;                  Generate loadable service vector for CJF$GETCJI.
0000  ;
0000  ;          V03-030 KBT0578          Keith B. Thompson         8-Aug-1983
0000  ;                  Add parameter to $FILESCAN
0000  ;
0000  ;          V03-029 RAS0178          Ron Schaefer              29-Jul-1983
0000  ;                  Add code to detect the AST/non-AST RMS FAB/RAB race
0000  ;                  condition where an RMS operation is initiated while
0000  ;                  the user FAB/RAB is still waiting for completion of
0000  ;                  previous operation.
0000  ;
0000  ;          V03-028 WMC0028          Wayne Cardoza             29-Jun-1983
0000  ;                  Add CJF services.
0000  ;
0000  ;          V03-027 WMC0027          Wayne Cardoza             23-Jun-1983
0000  ;                  Make old logical name services "all mode".
0000  ;                  Changes to image activator vectors.
0000  ;
0000  ;          V03-026 JWH0222          Jeffrey W. Horn            2-May-1983
0000  ;                  Add LDBSRV macro for vector definitions of loadable
0000  ;                  services.
0000  ;
0000  ;          V03-025 DMW4035          DMWalp                    26-May-1983
0000  ;                  Intergate new logical name structures.
0000  ;
0000  ;          V03-024 LMP0109          L. Mark Pilant,           28-Apr-1983  15:53
0000  ;                  Make $CHKPRO an EXEC mode system service to allow examination
0000  ;                  of various system data structures.
0000  ;
```

```
0000  ;       V03-024 RAS0147         Ron Schaefer            28-APR-1983
0000  ;               Add $FILESCAN. Add R8 and R9 to $SETPRN register mask.
0000  ;
0000  ;       V03-023 JLV0244         Jake VanNoy             27-APR-1983
0000  ;               Add $BRKTHRUW. Change $BRDCST to all mode service.
0000  ;               $BRDCST now uses $BRKTHRU to do real work.
0000  ;
0000  ;       V03-022 LMP0099         L. Mark Pilant,         13-Apr-1983  19:15
0000  ;               Add the $CHKPRO system service.
0000  ;
0000  ;       V03-021 ACG0319         Andrew C. Goldstein,    21-Mar-1983  13:51
0000  ;               Add $GRANTID and $REVOKID services
0000  ;
0000  ;       V03-020 JLV0234         Jake VanNoy              1-MAR-1983
0000  ;               Add $BRKTHRU service.
0000  ;
0000  ;       V03-019 RAS0120         Ron Schaefer            25-Feb-1983
0000  ;               Add support to echo SYS$INPUT to SYS$OUTPUT.
0000  ;               This involves examining the return code from RMS for $GET;
0000  ;               if the special status RMS$_ECHO (not returned to users)
0000  ;               is found, then create a RAB on the caller's stack and
0000  ;               execute a $PUT operation to echo the line.
0000  ;               A certain amount of RMS synchronization code was
0000  ;               shuffled around in order to make room for this.
0000  ;
0000  ;       V03-018 ACG0317         Andrew C. Goldstein,    22-Feb-1983  15:16
0000  ;               Fix off-by-one in kernel arg vector
0000  ;
0000  ;       V03-017 RSH0004         R. Scott Hanna          10-Feb-1983
0000  ;               Added $ASCTOID, $FINISH_RDB, and $IDTOASC to system service list
0000  ;
0000  ;       V03-016 RNG0016         Rod N. Gamache          1-Feb-1983
0000  ;               Added $GETLKI to system service list
0000  ;
0000  ;       V03-015 WMC0015         Wayne Cardoza           12-Jan-1983
0000  ;               Put back accidentally deleted space holder for RMS synchronization.
0000  ;
0000  ;       V03-014 DMW4023         DMWalp                  7-Jan-1983
0000  ;               Added $CRELNT, $CRELNM, $DELLNM and $TRNLNM
0000  ;
0000  ;       V03-013 KDM0033         Kathleen D. Morse       13-Dec-1982
0000  ;               Correct usage of an interlocked instruction to flush
0000  ;               the hardware cache queue.
0000  ;
0000  ;       V03-012 ROW0146         Ralph O. Weber          6-DEC-1982
0000  ;               Insert routine header comments for INHEXCP, CHECKARGLIST,
0000  ;               and EXE$CMODKRNLX (MPS$CMODKRNLX).  Move things around so
0000  ;               that EXE$CMODKRNL (MPS$CMODKRNL) header comments are near
0000  ;               EXE$CMODRKNL (MPS$CMODKRNL) and ASTEXIT comments are near
0000  ;               ASTEXIT.  Make basic kernal-mode .PSECT definition for Y$CMODK
0000  ;               or MP$CMOD1 immediately after executive mode code so that new
0000  ;               code can be inserted in a way that preserves routine headers,
0000  ;               conditional assembly, and .PSECT definitions.  Backout ROW145,
0000  ;               and in its place, correct conditional assembly of BGEQU 10$
0000  ;               after ACCVIO_RET so that it is assembled only for MPCMOD and
0000  ;               so that it is located before ACCVIO_RET.  Change PCB address
0000  ;               lookup at KERDSP in MPCMOD to use CTL$GL_PCB so that it works
```

```
0000     ;              correctly regardless of which processor executes it.
0000     ;
0000     ;      V03-011 ROW0145         Ralph O. Weber          29-NOV-1982
0000     ;              Move EXE$EXCPTN (and MPS$EXCPTN) to before ASTEXIT (or
0000     ;              MPS$ASTEXIT) in an attempt to make branch destinations in
0000     ;              EXE$CMODKRNL reach.
0000     ;
0000     ;      V03-010 KDM0030         Kathleen D. Morse       18-Nov-1982
0000     ;              Add logic to MPCMOD that allows the primary to execute
0000     ;              secondary-specific code, without turning into a secondary.
0000     ;
0000     ;      V03-009 MLJ0099         Martin L. Jack, 20-Oct-1982  19:42
0000     ;              Complete V03-002 by correcting mode and argument count of
0000     ;              $SNDJBC and removing temporary stubs.
0000     ;
0000     ;      V03-008 RIH0001         Richard I. Hustvedt     1-Jun-1982
0000     ;              Correct handling of AST queue by secondary processor to
0000     ;              avoid losing some AST notifications by incorrectly computing
0000     ;              PHD$B_ASTLVL.
0000     ;
0000     ;      V03-007 KDM0018         Kathleen D. Morse       30-Sep-1982
0000     ;              Add MPSWITCH logic to create a kernel system service
0000     ;              dispatcher for the secondary processor of an 11/782.
0000     ;
0000     ;      V03-006 STJ3028         Steven T. Jeffreys      26-Sep-1982
0000     ;              Added $ERAPAT system service vector.
0000     ;
0000     ;      V03-005 DWT0058         David Thiel             11-Aug-1982
0000     ;              Eliminate use of R2 while waiting for service
0000     ;              completion.
0000     ;
0000     ;      V03-004 JWH0001         Jeffrey W. Horn         26-Jul-1982
0000     ;              Add new RMS service, RMSRUHNDLR, an un-documented service
0000     ;              which acts as the Recovery Unit handler for RMS.
0000     ;
0000     ;      V03-003 PHL0102         Peter H. Lipman         16-Jul-1982
0000     ;              Fix new SYNCH logic to always return SS$_NORMAL,
0000     ;              not access IOSB if error from service, and return
0000     ;              error status from $SETEF if event flag cluster went away
0000     ;
0000     ;      V03-002 PHL0101         Peter H. Lipman         17-Jun-1982
0000     ;              Add $SYNCH system service and fix $QIOW and $ENQW to use the
0000     ;              new code for waiting for the combination of EFN and IOSB
0000     ;
0000     ;              Improve readability of conditionals.
0000     ;
0000     ;              Add $GETDVIW, $GETJPIW, $GETSYIW, $SNDJBC, $SNDJBCW, and
0000     ;              $UPDSECW.  All the waiting versions use common code.
0000     ;
0000     ;
0000     ;
0000     ; CHANGE MODE SYSTEM SERVICE DISPATCHER
0000     ;
0000     ; MACRO LIBRARY CALLS
0000     ;
0000
0000            $ACBDEF                         ;DEFINE AST CONTROL BLOCK OFFSETS
```

```
                    0000   244          $CHFDEF                                 ;DEFINE CONDITION HANDLING OFFSETS
                    0000   245          $ENQDEF                                 ;DEFINE ENQ SYSTEM SERVICE ARGS
                    0000   246          $GETDVIDEF                              ;DEFINE GETDVI SYSTEM SERVICE ARGS
                    0000   247          $GETJPIDEF                              ;DEFINE GETJPI SYSTEM SERVICE ARGS
                    0000   248          $GETLKIDEF                              ;DEFINE GETLKI SYSTEM SERVICE ARGS
                    0000   249          $GETSYIDEF                              ;DEFINE GETSYI SYSTEM SERVICE ARGS
                    0000   250          $IPLDEF                                 ;DEFINE INTERRUPT PRIORITY LEVELS
                    0000   254          $PCBDEF                                 ;DEFINE PCB OFFSETS
                    0000   255          $PHDDEF                                 ;DEFINE PHD OFFSETS
                    0000   256          $PRDEF                                  ;DEFINE PROCESSOR REGISTERS
                    0000   257          $PSLDEF                                 ;DEFINE PROCESSOR STATUS FIELDS
                    0000   258          $RABDEF                                 ;DEFINE RMS RAB FIELDS
                    0000   259          $RPBDEF                                 ;DEFINE REBOOT PARAMETER BLOCK
                    0000   260          $QIODEF                                 ;DEFINE QIO SYSTEM SERVICE ARGS
                    0000   261          $SGNDEF                                 ;DEFINE SYSGEN PARAMETERS
                    0000   262          $SNDJBCDEF                              ;DEFINE SNDJBC SYSTEM SERVICE ARGS
                    0000   263          $SSDEF                                  ;DEFINE SYSTEM STATUS VALUES
                    0000   264          $SYNCHDEF                               ;DEFINE SYNCH SYSTEM SERVICE ARGS
                    0000   265          $UPDSECDEF                              ;DEFINE UPDATE SECTION SYS SRV ARGS
                    0000   266  ;
                    0000   267  ; LOCAL EQUATES
                    0000   268  ;
           00000001 0000   269          CAT0 =          1@0
           00000080 0000   270          CAT7 =          1@7
           00000081 0000   271          DEF_MASK =      CAT0!CAT7       ;INHIBIT FOR 'ALL' AND 'NOT EXIT'
           00000080 0000   272          EXC_MASK =      CAT7            ;INHIBIT ONLY FOR 'ALL' CASE
                    0000   273  ;
                    0000   274  ; LOCAL MACROS
                    0000   275  ;
                    0000   276  ;       GSYSSRV - GENERATE SYSTEM SERVICE ENTRY VECTOR
                    0000   277  ;
                    0000   278  ;       GSYSSRV SRVNAME,MODE,NARG,REGISTERS,MASK,NOSYNC
                    0000   279  ;
                    0000   280  ;       WHERE:
                    0000   281  ;               SRVNAME - SERVICE NAME LESS ANY PREFIX (SYS$,EXE$,RMS$$)
                    0000   282  ;               MODE - MODE DESIGNATOR FOR SERVICE (K,E,ALL,R)
                    0000   283  ;               NARG - REQUIRED NUMBER OF ARGUMENTS
                    0000   284  ;               REGISTERS - REGISTER SAVE LIST
                    0000   285  ;               MASK - SERVICE INHIBIT MASK(BIT SET IN CAT INHIBITS)
                    0000   286  ;               NOSYNC - NON-ZERO IF RMS SYNCHRONIZATION CODE NOT TO BE INCLUDED
                    0000   287  ;
                    0000   288
                    0000   289          .MACRO  GSYSSRV,SRVNAME,MODE,NARG,REGS,MASK=DEF_MASK,NOSYNC
                    0000   290          .IF     NDF,RMSSWITCH
                    0000   291          .IF     DF,LIBSWITCH
                    0000   292          .PSECT  $$$0000,QUAD
                    0000   293          .IFF
                    0000   294          .PSECT  $$$000,QUAD
                    0000   295          .ENDC
                    0000   296          .ALIGN  QUAD
                    0000   297          .IF DF  LIBSWITCH
                    0000   298  SYS$'SRVNAME::
                    0000   299          .IFF
                    0000   300          .IF     NDF,MPSWITCH
                    0000   301          .WORD   ^M<REGS>
                    0000   302          SRVNAME'_MASK = ^M<REGS>
                    0000   303          .IFTF   ;MPSWITCH
```

```
0000                    .IF B   NOSYNC
0000                    SRV'MODE        SRVNAME,NARG,MASK
0000                    .IFF
0000                    SRV'MODE        SRVNAME,NARG,MASK,NOSYNC
0000                    .ENDC
0000                    .ENDC   ;MPSWITCH
0000                    .IFT
0000                    .BLKL   2
0000                    .ENDC
0000                    .IFF
0000                    SRV'MODE        SRVNAME,NARG,MASK
0000                    .ENDC
0000                    .ENDM   GSYSSRV
0000
0000            ;
0000            ;       GCOMPSRVB - GENERATE COMPOSITE SYSTEM SERVICE ENTRY VECTOR BEGIN
0000            ;
0000            ;       GCOMPSRVB  SRVNAME,REGISTER_MASK[,PREFIX]
0000            ;
0000            ;       WHERE:
0000            ;               SRVNAME - SERVICE NAME LESS ANY PREFIX (SYS$, EXE$)
0000            ;               REGISTER_MASK - SYMBOLIC REGISTER MASK, E.G QIO_MASK
0000            ;               PREFIX - IF SUPPLIED, THE PREFIX FOR THE SERVICE NAME.
0000            ;                        IF OMITTED, "SYS$" IS ASSUMED.
0000            ;
0000
0000                    .MACRO  GCOMPSRVB,SRVNAME,REGMSK,PREFIX=SYS$
0000                    .IF     NDF,MPSWITCH
0000                    .IF     NDF,RMSSWITCH
0000                    .IF     DF,LIBSWITCH
0000                    .PSECT  $$$0000,QUAD
0000                    .IFF
0000                    .PSECT  $$$000,QUAD
0000                    .ENDC
0000                    .ALIGN  QUAD
0000                    .IF DF  LIBSWITCH
0000                    .IIF    NOT_BLANK, <SRVNAME>,-
0000            'PREFIX'SRVNAME::
0000                    .IFF
0000                    .ENABL  LSB
0000            COMPSTRT=.
0000                    .IIF    NOT_BLANK, <REGMSK>,-
0000                    .WORD   <REGMSK>
0000                    .ENDC
0000                    .ENDC
0000                    .ENDC   ;MPSWITCH
0000                    .ENDM   GCOMPSRVB
0000
0000            ;
0000            ;       GCOMPSRVE - GENERATE COMPOSITE SYSTEM SERVICE ENTRY VECTOR END
0000            ;
0000            ;       GCOMPSRVE       QUADWORDS
0000            ;
0000            ;       WHERE:
0000            ;               QUADWORDS - NUMBER OF QUADWORDS TO RESERVE FOR VECTOR
0000            ;
0000            ;
```

```
0000            .MACRO  GCOMPSRVE,QUADS
0000            .IF     NDF,MPSWITCH
0000            .IF     NDF,RMSSWITCH
0000            .IF     DF,LIBSWITCH
0000            .BLKQ   QUADS
0000            .IFF
0000    COMPSIZE=.-COMPSTRT
0000            .IF     GE,QUADS*8-COMPSIZE
0000            .BLKB   QUADS*8-COMPSIZE
0000            .IFF
0000            .ERROR          ; VECTOR EXCEEDS ALLOCATED SIZE ;
0000            .ENDC
0000            .DSABL  LSB
0000            .ENDC
0000            .ENDC
0000            .ENDC   ;MPSWITCH
0000            .ENDM   GCOMPSRVE
0000
0000
0000    ;
0000    ;       SRVK - GENERATE ENTRY FOR KERNEL MODE SERVICE
0000    ;
0000    ;       SRVK    SRVNAME,NARG,MASK
0000    ;
0000
0000            .MACRO  SRVK,SRVNAME,NARG,MASK
0000            .IF     NDF,RMSSWITCH
0000            .IF     DF,MPSWITCH
0000    CMK$C_'SRVNAME==KCASCTR
0000            .IFF    ;MPSWITCH DEFINED
0000    CMK$C_'SRVNAME=KCASCTR
0000            CHMK    #SRVNAME
0000            RET
0000            .PSECT  Y$CMODKN,BYTE
0000            .=KCASCTR
0000            ASSUME NARG LE 127
0000            .BYTE   NARG
0000            .PSECT  Y$CMODKX,BYTE
0000            .=KCASCTR
0000            .BYTE   MASK
0000            .PSECT  Y$CMODK,BYTE
0000            .SIGNED_WORD    EXE$'SRVNAME-KCASE+2
0000            .IFTF   ;MPSWITCH
0000    SRVNAME=KCASCTR
0000    KCASCTR=KCASCTR+1
0000            .ENDC   ;MPSWITCH
0000            .ENDC
0000            .ENDM   SRVK
0000
0000    ;
0000    ;       SRVE - GENERATE ENTRY FOR EXECUTIVE MODE SERVICE
0000    ;
0000
0000            .MACRO  SRVE,SRVNAME,NARG,MASK
0000            .IF     NDF,MPSWITCH
0000            .IF     NDF,RMSSWITCH
0000    CME$C_'SRVNAME=ECASCTR
```

```
0000            CHME    #SRVNAME
0000            RET
0000            .PSECT  Y$CMODEN,BYTE
0000            .=ECASCTR
0000            ASSUME NARG LE 127
0000            .BYTE   NARG
0000            .PSECT  Y$CMODEX,BYTE
0000            .=ECASCTR
0000            .BYTE   MASK
0000            .PSECT  Y$CMODE,BYTE
0000            .SIGNED_WORD    EXE$'SRVNAME-ECASE+2
0000            .ENDC
0000    SRVNAME=ECASCTR
0000    ECASCTR=ECASCTR+1
0000            .ENDC   ;MPSWITCH
0000            .ENDM   SRVE
0000    ;
0000    ;
0000    ;     MACROS FOR GENERATING RMS SYSTEM VECTORS
0000    ;
0000            .MACRO  RMSSRV  SRVNAME NARG=1,REGS=<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>,-
0000                            MASK,NOSYNC=0
0000            GSYSSRV SRVNAME,R,NARG,<REGS>,MASK,NOSYNC
0000            .ENDM   RMSSRV
0000    ;
0000    ;   SRVR - GENERATE ENTRY FOR RMS SERVICE (EXEC MODE)
0000    ;
0000            .MACRO  SRVR    SRVNAME,NARG,MASK,NOSYNC
0000            .IF     NDF,MPSWITCH
0000            .IF     NDF,RMSSWITCH
0000    CME$C_'SRVNAME=RCASCTR
0000            CHME    #SRVNAME
0000            .IF EQ NOSYNC
0000            .IIF GT <.+2-RMSSYNC>-127,-
0000    RMSSYNC=RMSWBR                                  ;RESET BRANCH DESTINATION
0000    RMSWBR=.
0000            BRB     RMSSYNC
0000            .IFF
0000            RET
0000            .ENDC
0000            .PSECT  Y$CMODEN,BYTE
0000            .=RCASCTR
0000            ASSUME NARG LE 127
0000            .BYTE   NARG
0000            .PSECT  Y$CMODEX,BYTE
0000            .=RCASCTR
0000            .BYTE   MASK
0000            .IFF
0000            .PSECT  $$$RMSVEC,BYTE,NOWRT
0000            .SIGNED_WORD    RMS$'SRVNAME-RCASE+2
0000            .ENDC
0000    SRVNAME=RCASCTR
0000    RCASCTR=RCASCTR+1
0000            .ENDC   ;MPSWITCH
0000            .ENDM   SRVR
0000
0000    ;
```

```
0000  ;       SRVALL - GENERATE ENTRY FOR ALL MODE SERVICE
0000  ;
0000
0000          .MACRO  SRVALL,SRVNAME,NARG,MASK
0000          .IF     NDF,MPSWITCH
0000          .IF NDF,RMSSWITCH
0000          JMP     @#EXE$'SRVNAME+2
0000          .ENDC
0000          .ENDC   ;MPSWITCH
0000          .ENDM   SRVALL
0000
```

```
0000    487             .SBTTL  Macros for Loadable Services
0000    488
0000    489 ;
0000    490 ;           LDBSRV - Generate Loadable Service Vector
0000    491 ;
0000    492 ;           LDBSRV  PREFIX,SRVNAME,MODE,REGS,SYN_EFN,SYN_IOSB,ALT_CHMX
0000    493 ;
0000    494 ;           Where:
0000    495 ;                   PREFIX          - Prefix for system service vector entry point name
0000    496 ;                   SRVNAME         - Service name less any prefix (SYS$,CJF$, etc.)
0000    497 ;                   MODE            - Mode designator for service (K,E,ALL)
0000    498 ;                   REGS            - Register save list
0000    499 ;                   SYN_EFN         - Event flag argument number for $SYNCH
0000    500 ;                   SYN_IOSB        - IOSB argument number for $SYNCH
0000    501 ;                   ALT_CHMX        - Use same CHMx number as this service
0000    502 ;
0000    503
0000    504             .MACRO  LDBSRV,PREFIX,SRVNAME,MODE,REGS,SYN_EFN,SYN_IOSB,ALT_CHMX
0000    505             .IF NDF,RMSSWITCH
0000    506             .IF NDF,MPSWITCH
0000    507                 .IF DF,LIBSWITCH
0000    508                     .PSECT  $$$0000,QUAD
0000    509                     .ALIGN  QUAD
0000    510 PREFIX''SRVNAME::
0000    511                     .IF BLANK SYN_EFN
0000    512                         .BLKL    2
0000    513                     .IFF
0000    514                         .BLKL    4
0000    515                     .ENDC
0000    516                 .IFF
0000    517                     .PSECT  $$$000,QUAD
0000    518                     .ALIGN  QUAD
0000    519                     .WORD   ^M<REGS>
0000    520                     SRVNAME'_MASK = ^M<REGS>
0000    521                     LVEC_'MODE PREFIX,SRVNAME,SYN_EFN,SYN_IOSB,ALT_CHMX
0000    522                 .ENDC
0000    523             .ENDC   ; MPSWITCH
0000    524             .ENDC   ; RMSSWITCH
0000    525             .ENDM   LDBSRV
0000    526
0000    527 ;
0000    528 ;           LVEC_K  - Kernel Mode Loadable System Service Vector
0000    529 ;
0000    530 ;           LVEC_K  PREFIX,SERVICE,EFN,IOSB
0000    531 ;
0000    532
0000    533             .MACRO  LVEC_K,PREFIX,SERVICE,EFN,IOSB,ALT_CHMK
0000    534             .IF BLANK ALT_CHMK
0000    535                 CMK$C_'SERVICE = PREFIX'KCASCTR
0000    536             .IFF
0000    537                 CMK$C_'SERVICE = ALT_CHMK
0000    538             .ENDC
0000    539             CHMK #SERVICE
0000    540             .IF NOT_BLANK EFN
0000    541                 PUSHL       #EFN
0000    542                 PUSHL       #IOSB
0000    543                 JMP         @#EXE$LDB_SYNCH
```

```
                    0000    544         .IFF
                    0000    545             RET
                    0000    546         .ENDC
                    0000    547         .IF BLANK ALT_CHMK
                    0000    548             SERVICE = PREFIX'KCASCTR
                    0000    549             PREFIX'KCASCTR = PREFIX'KCASCTR + 1
                    0000    550         .IFF
                    0000    551             SERVICE = ALT_CHMK
                    0000    552         .ENDC
                    0000    553         .ENDM   LVEC_K
                    0000    554
                    0000    555 ;
                    0000    556 ;       LVEC_E  - Exec Mode Loadable System Service Vector
                    0000    557 ;
                    0000    558 ;       LVEC_E  PREFIX,SERVICE,EFN,IOSB
                    0000    559 ;
                    0000    560
                    0000    561         .MACRO  LVEC_E,PREFIX,SERVICE,EFN,IOSB,ALT_CHME
                    0000    562         .IF BLANK ALT_CHME
                    0000    563             CME$C_'SERVICE = PREFIX'ECASCTR
                    0000    564         .IFF
                    0000    565             CME$C_'SERVICE = ALT_CHME
                    0000    566         .ENDC
                    0000    567         CHME    #SERVICE
                    0000    568         .IF NOT BLANK EFN
                    0000    569             PUSHL       #EFN
                    0000    570             PUSHL       #IOSB
                    0000    571             JMP         @#EXE$LDB_SYNCH
                    0000    572         .IFF
                    0000    573             RET
                    0000    574         .ENDC
                    0000    575         RET
                    0000    576         .IF BLANK ALT_CHME
                    0000    577             SERVICE = PREFIX'ECASCTR
                    0000    578             PREFIX'ECASCTR = PREFIX'ECASCTR + 1
                    0000    579         .IFF
                    0000    580             SERVICE = ALT_CHME
                    0000    581         .ENDC
                    0000    582         .ENDM   LVEC_E
                    0000    583
                    0000    584 ;
                    0000    585 ;       LVEC_ALL - Mode of caller Loadable System Service Vector
                    0000    586 ;
                    0000    587 ;       LVEC_ALL   PREFIX,SERVICE,EFN,IOSB
                    0000    588 ;
                    0000    589         .MACRO  LVEC_ALL,PREFIX,SERVICE,EFN,IOSB,ALT_CHMK
                    0000    590         JMP     @#EXE$'SERVICE
                    0000    591         .IF NOT BLANK EFN
                    0000    592             .ERROR      ; SYNCH NOT ALLOWED FOR ALL-MODE SERVICES
                    0000    593         .ENDC
                    0000    594         .ENDM   LVEC_ALL
                    0000    595
                    0000    596
          00000000  0000    598 ECASCTR=0
```

```
0000     ;
0000     ;       DEFINE REMAINING SERVICES
0000     ;
0000
0000             GSYSSRV ADJSTK,K,3,-                    ;ADJUST OUTER MODE STACK POINTER
0000                     <R2,R3,R4,R5,R6>,-              ;REGISTERS R2-R6
0000                     EXC_MASK                        ;EXCEPTION MASK
0000             GSYSSRV ADJWSL,K,2,-                    ;ADJUST WORKING SET LIMIT
0000                     <R2,R3,R4,R5>                   ;REGISTERS R2-R5
0000             GSYSSRV ALCDNP,K,4,-                    ;ALLOCATE DIAGNOSTIC PAGE
0000                     <R2,R3,R4,R5,R6,R7>             ;REGISTERS R2-R7
0000             GSYSSRV ALLOC,K,4,-                     ;ALLOCATE DEVICE
0000                     <R2,R3,R4,R5,R6>                ;REGISTERS R2-R6
0000             GSYSSRV ASCEFC,K,4,-                    ;ASSOCIATE COMMON EVENT FLAG CLUSTER
0000                     <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000             GSYSSRV ASCTIM,ALL,3,-                  ;CONVERT TO ASCII TIME
0000                     <R2,R3,R4,R5,R6>                ;REGISTERS R2-R6
0000             GSYSSRV ASSIGN,K,4,-                    ;ASSIGN I/O CHANNEL
0000                     <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000             GSYSSRV BINTIM,ALL,2,-                  ;CONVERT TO BINARY TIME
0000                     <R2,R3,R4,R5,R6,R7,R8>          ;REGISTERS R2-R8
0000             GSYSSRV CANCEL,K,1,-                    ;CANCEL I/O ON CHANNEL
0000                     <R2,R3,R4,R5,R6,R7,R8>          ;REGISTERS R2-R8
0000             GSYSSRV CANTIM,K,2,-                    ;CANCEL TIMER REQUEST
0000                     <R2,R3,R4,R5>                   ;REGISTERS R2-R5
0000             GSYSSRV CANWAK,K,2,-                    ;CANCEL WAKE UP REQUESTS
0000                     <R2,R3,R4,R5>                   ;REGISTERS R2-R5
0000             GSYSSRV CRMPSC,K,12,-                   ;CREATE AND MAP SECTION
0000                     <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000             GSYSSRV CLRPAR,K,2,-                    ;CLEAR HARD PARITY ERROR
0000                     <R2,R3,R4,R5>                   ;REGISTERS R2-R5
0000             GSYSSRV CMEXEC,E,2,-                    ;CHANGE MODE TO EXECUTIVE
0000                     <R4>                            ;REGISTER R4
0000             GSYSSRV CMKRNL,K,2,-                    ;CHANGE MODE TO KERNEL
0000                     <R4>                            ;REGISTER R4
0000             GSYSSRV CLREF,K,1,-                     ;CLEAR EVENT FLAG
0000                     <R2,R3,R4,R5>                   ;REGISTERS R2-R5. SEE WAITFR COMMENTS.
0000             GSYSSRV CNTREG,K,4,-                    ;CONTRACT REGION
0000                     <R2,R3,R4,R5,R6,R7>             ;REGISTERS R2-R7
0000             GSYSSRV GETPTI,K,5,-                    ;GET PAGE TABLE INFORMATION
0000                     <R2,R3,R4,R5,R6,R7,R8,R9,R10> ;REGISTERS R2-R10
0000             GSYSSRV CRELOG,ALL,4,-                  ;CREATE LOGICAL NAME
0000                     <R2,R3,R4,R5,R6,R7,R8>          ;REGISTERS R2-R8
0000             GSYSSRV CREMBX,K,7,-                    ;CREATE MAILBOX
0000                     <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000             GSYSSRV CREPRC,K,12,-                   ;CREATE PROCESS
0000                     <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000             GSYSSRV CRETVA,K,3,-                    ;CREATE VIRTUAL ADDRESS
0000                     <R2,R3,R4,R5,R6,R7,R8>,-        ;REGISTERS R2-R8
0000                     EXC_MASK                        ;EXCEPTION MASK
0000             GSYSSRV DACEFC,K,1,-                    ;DISASSOCIATE EVENT FLAG CLUSTER
0000                     <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000             GSYSSRV DALLOC,K,2,-                    ;DEALLOCATE DEVICE
0000                     <R2,R3,R4,R5,R8>                ;REGISTERS R2-R5,R8
0000             GSYSSRV DASSGN,K,1,-                    ;DEASSIGN I/O CHANNEL
0000                     <R2,R3,R4,R5,R6,R7,R8>          ;REGISTERS R2-R8
0000             GSYSSRV DCLAST,K,3,-                    ;DECLARE AST SYSTEM SERVICE
```

```
SYS
Sym

RMS
RMS
RMS
RMS
RMS
RMS
RMS
RMS
RMS
RMS
RMS
RMS
RMS
RMS
RMS
RMS
RMS
RMS
RMS
RMS
RMS
RMS
RMS
RMS
RMS
RMS
RMS
RUF
SEA
SET
SET
SND
SND
SND
SND
SND
SND
SND
SND
SND
SND
SND
SND
SPA
SSV
SYN
SYN
SYN
TRU
UPD
UPD
UPD
UPD
UPD
UPD
UPD
UPD
```

```
0000                    <R2,R3,R4,R5>                   ;REGISTERS R2-R5
0000            GSYSSRV DCLEXH,K,1,-                    ;DECLARE EXIT HANDLER
0000                    <R2,R3,R4>                      ;REGISTERS R2-R4
0000            GSYSSRV DELLOG,ALL,3,-                  ;DELETE LOGICAL NAME
0000                    <R2,R3,R4,R5,R6,R7,R8>          ;REGISTERS R2-R8
0000            GSYSSRV DELMBX,K,1,-                    ;DELETE MAILBOX
0000                    <R2,R3,R4,R5>                   ;REGISTERS R2-R5
0000            GSYSSRV DELPRC,K,2,-                    ;DELETE PROCESS
0000                    <R2,R3,R4,R5,R6,R7>             ;REGISTERS R2-R5
0000            GSYSSRV DELTVA,K,3,-                    ;DELETE VIRTUAL ADDRESS
0000                    <R2,R3,R4,R5,R6,R7>,-           ;REGISTERS R2-R7
0000                    EXC_MASK                        ;EXCEPTION MASK
0000            GSYSSRV DGBLSC,K,3,-                    ;DELETE GLOBAL SECTION
0000                    <R2,R3,R4,R5,R6,R7,R8,R9,R10> ;REGISTERS R2-R10
0000            GSYSSRV DLCDNP,K,2,-                    ;DEALLOCATE DIAGNOSTIC PAGE
0000                    <R2,R3,R4,R5,R6,R7>             ;REGISTERS R2-R7
0000            GSYSSRV DLCEFC,K,1,-                    ;DELETE COMMON EVENT CLUSTER
0000                    <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000            GSYSSRV UPDSEC,K,8,-                    ;UPDATE SECTION FILE
0000                    <R2,R3,R4,R5,R6,R7,R8>          ;R2-R8
0000            GSYSSRV SNDERR,K,1,-                    ;SEND MSG TO ERROR LOGGER
0000                    <R2,R3,R4,R5>                   ;REGISTERS R2-R5
0000            GSYSSRV EXIT,K,1,-                      ;IMAGE EXIT
0000                    <R4>,0                          ;REGISTER R4, ALWAYS ALLOWED!
0000            GSYSSRV EXPREG,K,4,-                    ;EXPAND PROGRAM REGION
0000                    <R2,R3,R4,R5,R6,R7,R8>          ;REGISTERS R2-R8
0000            GSYSSRV FAO,ALL,0,-                     ;FORMAT ASCII OUTPUT
0000                    <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
0000            GSYSSRV FAOL,ALL,0,-                    ;FORMAT ASCII OUTPUT WITH VALUE LIST
0000                    <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000            GSYSSRV FORCEX,K,3,-                    ;FORCE EXIT
0000                    <R2,R3,R4,R5>                   ;REGISTERS R2-R5
0000            GSYSSRV IMGSTA,ALL,6,-                  ;IMAGE STARTUP
0000                    <>                              ;REGISTERS NONE
0000            GSYSSRV SNDJBC,E,7,-                    ;SEND TO JOB CONTROLLER
0000                    <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000            GSYSSRV GETTIM,E,1,-                    ;GET TIME
0000                    <>                              ;NO REGISTERS
0000            GCOMPSRVB UPDSECW,-                     ;UPDATE SECTION AND WAIT
0000                    <UPDSEC_MASK ! GETJPI_SYNCH_MASK>
0000            GCOMPSRVE       1
0000            GSYSSRV HIBER,K,0,-                     ;HIBERNATE
0000                    <R2,R3,R4,R5>                   ;REGISTERS R2-R5
0000            GSYSSRV IMGACT,E,8,-                    ;IMAGE ACTIVATION
0000                    <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000            GSYSSRV LCKPAG,K,3,-                    ;LOCK PAGE IN MEMORY
0000                    <R2,R3,R4,R5,R6,R7,R8>          ;REGISTERS R2-R8
0000            GSYSSRV LKWSET,K,3,-                    ;LOCK PAGES IN WORKING SET
0000                    <R2,R3,R4,R5,R6,R7,R8>          ;REGISTERS R2-R8
0000            GSYSSRV MGBLSC,K,7,-                    ;MAP GLOBAL SECTION
0000                    <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000            GSYSSRV PURGWS,K,1,-                    ;PURGE WORKING SET
0000                    <R2,R3,R4,R5,R6,R7,R8>          ;R2-R8
0000            GSYSSRV NUMTIM,E,2,-                    ;CONVERT TIME TO NUMERIC
0000                    <R2,R3,R4,R5,R6,R7>             ;REGISTERS R2-R7
0000            GSYSSRV SNDOPR,E,2,-                    ;SEND MSG TO OPERATOR
0000                    <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>       ;REGISTERS R2-R11
```

```
                      0000  1334          GSYSSRV QIO,K,12,-                  ;QUEUE I/O REQUEST
                      0000  1335                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
                      0000  1336          GSYSSRV READEF,K,2,-                ;READ EVENT FLAG
                      0000  1337                  <R2,R3,R4,R5>               ;REGISTERS R2-R5
                      0000  1338          GSYSSRV RESUME,K,2,-                ;RESUME PROCESS
                      0000  1339                  <R2,R3,R4,R5>               ;REGISTERS R2-R5
                      0000  1340          GSYSSRV RUNDWN,K,1,-                ;RUNDOWN
                      0000  1341                  <R2,R3,R4,R5,R6,R7>         ;REGISTERS R2-R7
                      0000  1342          GSYSSRV SNDSMB,E,2,-                ;SEND MSG TO SYMBIONT MANAGER
                      0000  1343                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>         ;REGISTERS R2-R11
                      0000  1344          GSYSSRV SCHDWK,K,4,-                ;SCHEDULE WAKEUP
                      0000  1345                  <R2,R3,R4,R5,R6,R7,R8,R9>  ;REGISTERS R2-R9
                      0000  1346          GSYSSRV SETAST,K,1,-                ;SET AST ENABLE SERVICE
                      0000  1347                  <R2,R3,R4,R5>               ;REGISTERS R2-R5
                      0000  1348          GSYSSRV SETEF,K,1,-                 ;SET EVENT FLAG
                      0000  1349                  <R2,R3,R4,R5>               ;REGISTERS R2-R5. SEE WAITFR COMMENTS.
                      0000  1350          GSYSSRV SETEXV,K,4,-                ;SET EXCEPTION VECTOR
                      0000  1351                  <R2,R3,R4,R5>               ;REGISTERS R2-R5
                      0000  1352          GSYSSRV SETPRN,K,1,-                ;SET PROCESS  NAME
                      0000  1353                  <R2,R3,R4,R5,R6,R7,R8,R9>  ;REGISTERS R2-R9
                      0000  1354          GSYSSRV SETPRA,K,2,-                ;SET POWER RECOVERY AST
                      0000  1355                  <R2,R3,R4,R5>               ;REGISTERS R2-R5
                      0000  1356          GSYSSRV SETIMR,K,4,-                ;SET TIMER
                      0000  1357                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
                      0000  1358          GSYSSRV SETPRI,K,4,-                ;SET PROCESS PRIORITY
                      0000  1359                  <R2,R3,R4,R5>               ;REGISTERS R2-R5
                      0000  1360          GSYSSRV SETPRT,K,5,-                ;SET PAGE PROTECTION
                      0000  1361                  <R2,R3,R4,R5,R6,R7,R8,R9>  ;REGISTERS R2-R9
                      0000  1362          GSYSSRV SETRWM,K,1,-                ;SET RESOURCE WAIT MODE
                      0000  1363                  <R4>                        ;REGISTER R4
                      0000  1364          GSYSSRV SETSFM,K,1,-                ;SET SYSTEM SERVICE FAILURE MODE
                      0000  1365                  <R4>,EXC_MASK               ;REGISTER R4, AND EXECPTION MASK
                      0000  1366          GSYSSRV SETSWM,K,1,-                ;SET PROCESS SWAP MODE
                      0000  1367                  <R4>                        ;REGISTER R4
                      0000  1368          GSYSSRV SUSPND,K,2,-                ;SUSPEND PROCESS
                      0000  1369                  <R2,R3,R4,R5>               ;REGISTERS R2-R5
                      0000  1370          GSYSSRV TRNLOG,ALL,6,-              ;TRANSLATE LOGICAL NAME
                      0000  1371                  <R2,R3,R4,R5,R6,R7,R8>      ;REGISTERS R2-R8
                      0000  1372          GSYSSRV ULKPAG,K,3,-                ;UNLOCK PAGE FROM MEMORY
                      0000  1373                  <R2,R3,R4,R5,R6,R7,R8>      ;REGISTERS R2-R8
                      0000  1374          GSYSSRV ULWSET,K,3,-                ;UNLOCK PAGES FROM WORKING SET
                      0000  1375                  <R2,R3,R4,R5,R6,R7,R8>      ;REGISTERS R2-R8
                      0000  1376          GSYSSRV UNWIND,ALL,2,-              ;UNWIND PROCEDURE CALL STACK
                      0000  1377                  <R2,R3,R4,R5>               ;REGISTERS R2-R5
                      0000  1378          GSYSSRV WAITFR,K,1,-                ;WAIT FOR EVENT FLAG
                      0000  1379                  <R2,R3,R4,R5,R6>            ;REGISTERS R2-R6.  IF R8 IS EVER USED
                      0000  1380                                              ;THE RMS SYCHRONIZATION CODE MUST BE
                      0000  1381                                              ;MODIFIED TO SAVE IT ALSO.
                      0000  1382          GSYSSRV WAKE,K,2,-                  ;WAKE PROCESS
                      0000  1383                  <R2,R3,R4,R5>               ;REGISTERS R2-R5
                      0000  1384          GSYSSRV WFLAND,K,2,-                ;WAIT FOR LOGICAL AND OF EVENT FLAGS
                      0000  1385                  <R2,R3,R4,R5,R6>            ;REGISTERS R2-R6
                      0000  1386          GSYSSRV WFLOR,K,2,-                 ;WAIT FOR LOGICAL OR OF EVENT FLAGS
                      0000  1387                  <R2,R3,R4,R5,R6>            ;REGISTERS R2-R5
                      0000  1388          GSYSSRV BRDCST,ALL,2,-              ;BROADCAST TO TERMINALS
                      0000  1389                  <R2,R3,R4,R5,R6>            ;REGISTERS R2-R6
                      0000  1390          GSYSSRV DCLCMH,K,3,-                ;DECLARE CHANGE MODE HANDLER
```

```
0000  1391          <R4>                              ;SAVE R4
0000  1392  GSYSSRV SETPFM,K,4,-                      ;SET PAGE FAULT MONITORING
0000  1393          <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1394  GSYSSRV GETMSG,ALL,5,-                    ;GET MESSAGE
0000  1395          <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1396  GSYSSRV DERLMB,K,1,-                     ;DECLARE ERROR LOG MAILBOX
0000  1397          <R2,R3,R4,R5>                     ;REGISTERS R2-R5
0000  1398  GSYSSRV CANEXH,K,1,-                     ;CANCEL EXIT HANDLER
0000  1399          <R2,R3,R4,R5>                     ;REGISTERS R2-R5
0000  1400  GSYSSRV GETCHN,K,5,-                     ;GET CHANNEL INFORMATION
0000  1401          <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1402  GSYSSRV GETDEV,K,5,-                     ;GET DEVICE INFORMATION
0000  1403          <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1404  GSYSSRV GETJPI,K,7,-                     ;GET JOB PROCESS INFORMATION
0000  1405          <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1406  GSYSSRV PUTMSG,ALL,3,-                   ;PUT FORMATTED ERROR MESSAGE
0000  1407          <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1408  GSYSSRV EXCMSG,ALL,2,-                   ;OUTPUT EXCEPTION SUMMARY MESSAGE
0000  1409          <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1410  GSYSSRV SNDACC,E,2,-                     ;SEND MSG TO ACOUNTING MANAGER
0000  1411          <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1412  GSYSSRV SETIME,K,1,-                     ;SET SYSTEM TIME
0000  1413          <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000  1414  GSYSSRV SETPRV,K,4,-                     ;SET PRIVILEGES
0000  1415          <R2,R3,R4,R5,R6,R7,R8>            ;REGISTERS R2-R8
```

```
0000  ;
0000  ;       SPECIAL VECTORS FOR AST DELIVERY AND CLEARING
0000  ;
0000  ;       SYS$CLRAST CLEARS THE CURRENTLY ACTIVE AST STATUS
0000  ;
0000  ;       SYS$GL_ASTRET CONTAINS THE VALUE OF THE RETURN ADDRESS FROM
0000  ;       THE CALL INSTRUCTION USED TO DISPATCH AN AST. THIS VALUE CAN
0000  ;       BE USED WHEN SEARCHING UP THE STACK FOR THE AST CALL FRAME.
0000  ;
0000  ;
0000  ; NOTE THAT THE CODE IN PSECT $$$000 AT THIS POINT CANNOT EXCEED 320 (HEX)
0000  ; WITHOUT MODIFYING THE RMS SYNCHRONIZATION CODE WHICH PRECEDES THE RMS
0000  ; VECTORS WHICH CANNOT BE MOVED.
0000  ;
0000
```

```
                          0000  1478 ;
                          0000  1479 ; Set up the base for the RMS service codes. We leave a hole so that
                          0000  1480 ; other exec mode system services can be defined later in this module.
                          0000  1481 ; The hole is defined by the offset between ECASCTR and RCASCTR; it
                          0000  1482 ; is checked with an ASSUME at the end of all service definitions.
                          0000  1483 ;
                00000012  0000  1485 RCASCTR=ECASCTR+10
                          0000  1487
                          0000  1489 ;
                          0000  1490 ;  CASE DISPATCHER FOR RMS SERVICES
                          0000  1491 ;
                          0000  1492 ;       RO HAS SERVICE DISPATCH CODE.
                          0000  1493 ;       IF IN RANGE DISPATCHES TO APPROPRIATE RMS SERVICE,
                          0000  1494 ;       ELSE SIMPLY DOES AN RSB
                          0000  1495 ;
                      00000000  1496        .PSECT  $$$RMSVEC,BYTE,NOWRT     ;MUST BE FIRST PSECT IN RMS
                          0000  1497 RMS$DISPATCH:                          ;MUST BE FIRST CODE IN FIRST RMS PSECT
    22'  12'  50   AF     0000  1498        CASEW   R0,S^#RCASMIN,S^#RCASMAX
                          0004  1499
                          0004  1500 RCASE:
                00000012  0004  1503 RCASMIN=RCASCTR
                          0004  1629 ;
                          0004  1630 ;  HIGH USE RECORD OPERATIONS
                          0004  1631 ;
                          0004  1632        RMSSRV  DELETE           ;DELETE A RECORD
                          0006  1633        .NLIST CND
                          0006  1634        RMSSRV  FIND             ;FIND RECORD
                          0008  1635        RMSSRV  FREE             ;RELEASE LOCK ON ALL RECORDS
                          000A  1636        RMSSRV  GET              ;GET A RECORD
                          000C  1637        RMSSRV  PUT              ;PUT A RECORD
                          000E  1638        RMSSRV  READ             ;READ A BLOCK
                          0010  1639        RMSSRV  RELEASE          ;RELEASE LOCK ON NAMED RECORD
                          0012  1640        RMSSRV  UPDATE           ;REWRITE EXISTING RECORD
                          0014  1646        RMSSRV  WAIT             ;STALL FOR RECORD OPERATION COMPLETE
                          0016  1652        RMSSRV  WRITE            ;WRITE BLOCK
                          0018  1653 ;
                          0018  1654 ;  LOWER USAGE OPERATIONS
                          0018  1655 ;
                          0018  1656        RMSSRV  CLOSE            ;CLOSE FILE
                          001A  1657        RMSSRV  CONNECT          ;CONNECT RAB
                          001C  1658        RMSSRV  CREATE           ;CREATE FILE
                          001E  1659        RMSSRV  DISCONNECT       ;DISCONNECT RAB
                          0020  1660        RMSSRV  DISPLAY          ;DISPLAY FILE INFORMATION
                          0022  1661        RMSSRV  ERASE            ;ERASE (DELETE) FILE
                          0024  1662        RMSSRV  EXTEND           ;EXTEND FILE ALLOCATION
                          0026  1663        RMSSRV  FLUSH            ;FINISH I/O ACTIVITY FOR STREAM
                          0028  1664        RMSSRV  MODIFY           ;MODIFY FILE ATTRIBUTES
                          002A  1665        RMSSRV  NXTVOL           ;NEXT VOLUME
                          002C  1666        RMSSRV  OPEN             ;OPEN FILE
                          002E  1667        RMSSRV  REWIND           ;REWIND FILE
                          0030  1668        RMSSRV  SPACE            ;POSITION FOR TRANSFER
                          0032  1669        RMSSRV  TRUNCATE         ;TRUNCATE FILE
                          0034  1670        RMSSRV  ENTER            ;ENTER FILENAME INTO DIRECTORY
                          0036  1671        RMSSRV  PARSE            ;PARSE FILENAME SPECIFICATION
                          0038  1672        RMSSRV  REMOVE           ;REMOVE FILENAME FROM DIRECTORY
                          003A  1673        RMSSRV  RENAME,NARG=4    ;RENAME A FILE
                          003C  1674        RMSSRV  SEARCH           ;SEARCH A FILE DIRECTORY
```

```
003E         RMSSRV  SETDDIR,NARG=3,NOSYNC=1
0040                                 ;SET DEFAULT DIRECTORY STRING
0040         RMSSRV  SETDFPROT,REGS=<R2,R3>,NARG=2,NOSYNC=1
0042                                 ;SET DEFAULT FILE PROTECTION MASK
0042         RMSSRV  SSVEXC,REGS=<>,NOSYNC=1
0044                                 ;GENERATE SYS SERV EXCEPTION
0044         RMSSRV  RMSRUNDWN,NARG=2,NOSYNC=1
0046                                 ;PERFORM RUNDOWN ON RMS FILES
0046         RMSSRV  RMSRUHNDLR,NARG=5,NOSYNC=1
0048                                 ;RMS Recovery Unit Handler
0048         RMSSRV  FILESCAN,NARG=3,NOSYNC=1
004A                                 ;Perform syntax check for file specs
004A   ;
004A   ; ADD NEW RMS SERVICES IN FRONT OF THIS CODE!
004A   ;
004A   ; Now we add special non-vector code.  Because of the CASE instruction
004A   ; used at the front of RMS, this code (and any future additional code)
004A   ; must be the last element of the RMS area.
004A   ;
004A
004A         GCOMPSRVB               ;Helper branch to error processing
004A         GCOMPSRVE       1
004A
004A
```

```
004A            .SBTTL  REGION 2 OF SYS. SERV. VECTOR DEFINITIONS
004A
004A    ;
004A    ; Note: Service codes for exec mode services in this region are
004A    ; reserved by the offset defined above between RCASCTR and ECASCTR.
004A    ; If the ASSUME at the end of this section breaks, the offset must
004A    ; be increased.
004A    ;
004A
004A            GSYSSRV ENQ,K,11,-              ; ENQUEUE
004A                    <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
004A            GSYSSRV DEQ,K,4,-               ; DEQUEUE
004A                    <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
004A            GCOMPSRVB ENQW,-                ; ENQUEUE AND WAIT
004A                    <ENQ_MASK ! WAITFR_MASK ! CLREF_MASK ! SETEF_MASK>
004A            GCOMPSRVE       3               ; RESERVE 3 QUADWORDS FOR VECTOR
004A            GSYSSRV SETSSF,K,1,-            ; SET SYSTEM SERVICE FILTER MASK
004A                    <R4>                    ; REGISTER R4
004A            GSYSSRV SETSTK,K,3,-            ; SET STACK LIMITS
004A                    <R2,R3,R4>              ; REGISTERS R2,R3,R4
004A            GSYSSRV GETSYI,K,7,-            ; GET SYSTEM INFORMATION
004A                    <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
004A            GSYSSRV IMGFIX,ALL,0,-          ; IMAGE ADDRESS RELOCATION FIXUP
004A                    <R2,R3,R4,R5>           ; REGISTERS R2-R5
004A            GCOMPSRVB       IMGFIX_2,-      ; *********** TEMP ***********
004A                    <0>
004A            GCOMPSRVE       1               ; *********** TEMP ***********
004A            GSYSSRV GETDVI,K,8,-            ; GET DEVICE AND VOLUME INFORMATION
004A                    <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
004A            GCOMPSRVB GETDVIW,-             ; GET DEVICE INFORMATION AND WAIT
004A                    <GETDVI_MASK ! GETJPI_SYNCH_MASK>
004A            GCOMPSRVE       1
004A            GCOMPSRVB GETJPIW,-             ; GET JOB/PROCESS INFORMATION AND WAIT
004A                    <GETJPI_MASK ! GETJPI_SYNCH_MASK>
004A            GCOMPSRVE       2
004A            GCOMPSRVB GETSYIW,-             ; GET SYSTEM INFORMATION AND WAIT
004A                    <GETSYI_MASK ! GETJPI_SYNCH_MASK>
004A            GCOMPSRVE       1
004A            GCOMPSRVB SNDJBCW,-             ; SEND TO JOB CONTROLLER AND WAIT
004A                    <SNDJBC_MASK ! GETJPI_SYNCH_MASK>
004A            GCOMPSRVE       1
004A            GCOMPSRVB SYNCH,-               ; SYNCHRONIZE EFN AND IOSB
004A                    <WAITFR_MASK ! CLREF_MASK ! SETEF_MASK>
004A            GCOMPSRVE       6               ; RESERVE 6 QUADWORDS FOR VECTOR
004A            GSYSSRV ERAPAT,K,3,-            ; GENERATE A SECURITY  ERASE PATTERN
004A                    <R4>                    ; SAVE R4
004A            GSYSSRV CRELNT,K,8,-            ; CREATE LOGICAL NAME TABLE
004A                    <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
004A            GSYSSRV CRELNM,K,5,-            ; CREATE LOGICAL NAME
004A                    <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
004A            GSYSSRV DELLNM,K,3,-            ; DELETE LOGICAL NAME
004A                    <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
004A            GSYSSRV TRNLNM,K,5,-            ; TRANSLATE LOGICAL NAME
004A                    <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
004A            GSYSSRV GETLKI,K,7,-            ; GET LOCK INFORMATION
004A                    <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
004A            GCOMPSRVB GETLKIW,-             ; GET LOCK INFORMATION AND WAIT
```

```
                 004A  1875                <GETLKI_MASK ! WAITFR_MASK ! CLREF_MASK ! SETEF_MASK>
                 004A  1887        GCOMPSRVE      2                 ; RESERVE 2 QUADWORDS FOR VECTOR
                 004A  1888
                 004A  1889        GSYSSRV ASCTOID,E,3,-            ;ASCII TO IDENTIFIER CONVERSION
                 004A  1890                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
                 004A  1891        GSYSSRV FINISH_RDB,E,1,-         ;FINISH RDB CONTEXT STREAM
                 004A  1892                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
                 004A  1893        GSYSSRV IDTOASC,E,6,-            ;IDENTIFIER TO ASCII CONVERSION
                 004A  1894                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
                 004A  1895        GSYSSRV BRKTHRU,K,11,-           ;BREAK THROUGH WRITES
                 004A  1896                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
                 004A  1897        GSYSSRV GRANTID,ALL,5,-          ;GRANT IDENTIFIER TO PROCESS
                 004A  1898                <R2,R3>                  ;REGISTERS R2-R3
                 004A  1899        GSYSSRV REVOKID,ALL,5,-          ;REVOKE IDENTIFIER FROM PROCESS
                 004A  1900                <R2,R3>                  ;REGISTERS R2-R3
                 004A  1901        GSYSSRV CHKPRO,K,1,-             ;GENERAL PROTECTION CHECK ROUTINE
                 004A  1902                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
                 004A  1903        GCOMPSRVB BRKTHRUW,-             ; BREAK THOUGH WRITE AND WAIT
                 004A  1904                <BRKTHRU_MASK ! GETJPI_SYNCH_MASK>
                 004A  1913        GCOMPSRVE      2
                 004A  1914        GSYSSRV GETQUI,E,7,-             ;GET QUEUE INFORMATION
                 004A  1915                <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
                 004A  1916        GCOMPSRVB GETQUIW,-              ;GET QUEUE INFORMATION AND WAIT
                 004A  1917                <GETQUI_MASK ! GETJPI_SYNCH_MASK>
                 004A  1926        GCOMPSRVE      2
                 004A  1927
                 004A  1928 ;
       00004028  004A  1929        CJF$KCASCTR = 16424
                 004A  1930 ;
                 004A  1931        LDBSRV  CJF$, ALLJDR,      K,   <R4>
                 004A  1932        LDBSRV  CJF$, ASSJNL,      K,   <R4>
                 004A  1933        LDBSRV  CJF$, CONUIC,      K,   <R4>
                 004A  1934        LDBSRV  CJF$, CREJNL,      K,   <R4>
                 004A  1935        LDBSRV  CJF$, DEALJDR,     K,   <R4>
                 004A  1936        LDBSRV  CJF$, DEASJNL,     ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                 004A  1937        LDBSRV  CJF$, DEASJNL_INT, K,   <R4>
                 004A  1938        LDBSRV  CJF$, DELJNL,      K,   <R4>
                 004A  1939        LDBSRV  CJF$, DMTJMD,      K,   <R4>
                 004A  1940        LDBSRV  CJF$, DSPJNL,      K,   <R4>
                 004A  1941        LDBSRV  CJF$, GETJNL,      K,   <R4>
                 004A  1942        LDBSRV  CJF$, GETRUI,      K,   <R4>
                 004A  1943        LDBSRV  CJF$, MODFLT,      K,   <R4>
                 004A  1944        LDBSRV  CJF$, POSJNL,      K,   <R4>
                 004A  1945        LDBSRV  CJF$, READJNL,     K,   <R4>
                 004A  1946        LDBSRV  CJF$, RECOVER,     K,   <R4>
                 004A  1947        LDBSRV  CJF$, MNTJMD,      K,   <R4>
                 004A  1948        LDBSRV  CJF$, CRENWV,      K,   <R4>
                 004A  1949        LDBSRV  CJF$, CONJNLF,     K,   <R4>
                 004A  1950        LDBSRV  CJF$, DCNJNLF,     K,   <R4>
                 004A  1951        LDBSRV  CJF$, FORCEJNL,    ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                 004A  1952        LDBSRV  CJF$, FORCEJNLW,   ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                 004A  1953        LDBSRV  CJF$, WRITEJNL,    ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                 004A  1954        LDBSRV  CJF$, WRITEJNLW,   ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                 004A  1955        LDBSRV  CJF$, GETCJI,      K,   <R4>
                 004A  1956        LDBSRV  CJF$, DMTJMDW,     K,   <R4>, 4, 5, DMTJMD
                 004A  1957        LDBSRV  CJF$, MODFLTW,     K,   <R4>, 4, 5, MODFLT
                 004A  1958        LDBSRV  CJF$, POSJNLW,     K,   <R4>, 4, 5, POSJNL
```

```
          004A          LDBSRV  CJF$, READJNLW,    K,   <R4>, 4, 5, READJNL
          004A          LDBSRV  CJF$, RECOVERW,    K,   <R4>, 5, 6, RECOVER
          004A
          004A    ;
00004010  004A          RUF$KCASCTR = 16400
          004A    ;
          004A          LDBSRV  RUF$,  REENTERRU,    K,  <R2,R3,R4,R5,R6>
          004A          LDBSRV  RUF$,  STARTRU,      K,  <R2,R3,R4,R5,R6>
          004A          LDBSRV  RUF$,  PHASE1,       K,  <R2,R3,R4,R5,R6>
          004A          LDBSRV  RUF$,  PHASE2,       K,  <R2,R3,R4,R5,R6>
          004A          LDBSRV  RUF$,  CANCELRU,     K,  <R2,R3,R4,R5,R6>
          004A          LDBSRV  RUF$,  MARKPOINTRU,  K,  <R2,R3,R4,R5,R6>
          004A          LDBSRV  RUF$,  RESETRU,      K,  <R2,R3,R4,R5,R6>
          004A          LDBSRV  RUF$,  DCLRUH,       K,  <R2,R3,R4,R5,R6>
          004A          LDBSRV  RUF$,  CANRUH,       K,  <R2,R3,R4,R5,R6>
          004A          LDBSRV  RUF$,  RUSTATUS,     K,  <R2,R3,R4,R5,R6>
          004A    ;
          004A    ; End Recovery Unit consists of a two-phase commit, so we call each
          004A    ; phase separately.
          004A    ;
          004A          GCOMPSRVB ENDRU, <PHASE1_MASK ! PHASE2_MASK>, RUF$ ; End Recovery Unit
          004A          GCOMPSRVE      2
          004A          GSYSSRV MTACCESS,K,6,-          ;Mag tape installation specific access routi
          004A                  <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
          004A
          004A    ;
          004A    ; End of system service vector definitions. New system services are
          004A    ; to be added at this point.
          004A    ;
          004A          ASSUME  RCASMIN GE ECASCTR      ;Exec service codes must not collide with RM
          004A
```

```
                    00000022  004A  2273 RCASMAX=RCASCTR-<1+RCASMIN>
                      0000004A  2278         .PSECT  $$$RMSVEC,BYTE,NOWRT
                          05  004A  2279         RSB                             ;NOT AN RMS EXEC MODE SERVICE
                              004B  2280 ;
                              004B  2281 ;  SERVICE TO MERELY MOVE RMS STATUS CODE IN R2 TO R0 AND RET,
                              004B  2282 ;  THUS GENERATING A SYSTEM SERVICE FAILURE EXCEPTION IF ENABLED
                              004B  2283 ;
                    00000049  004B  2284 RMS$SSVEXC=.-2
              50   52   D0  004B  2285         MOVL    R2,R0                   ;MOVE STATUS CODE TO R0
                          04  004E  2286         RET                             ;AND LET RET DO THE REST
```

```
        004F  2345          .END
```

```
$$ARGS                          = 00000008                  GETLKI$_IOSB                    = 00000010
$$T1                            = 00000024                  GETLKI$_ITMLST                  = 0000000C
ASCTOID                         = 00000008                  GETLKI$_LKIDADR                 = 00000008
CAT0                            = 00000001                  GETLKI$_NARGS                   = 00000007
CAT7                            = 00000080                  GETLKI$_RESERVED                = 0000001C
CJF$KCASCTR                     = 00004028                  GETQUI                          = 0000000B
CLOSE                           = 0000001C                  GETSYI$_ASTADR                  = 00000018
CMEXEC                          = 00000000                  GETSYI$_ASTPRM                  = 0000001C
CONNECT                         = 0000001D                  GETSYI$_CSIDADR                 = 00000008
CREATE                          = 0000001E                  GETSYI$_EFN                     = 00000004
DEF_MASK                        = 00000081                  GETSYI$_IOSB                    = 00000014
DELETE                          = 00000012                  GETSYI$_ITMLST                  = 00000010
DISCONNECT                      = 0000001F                  GETSYI$_NARGS                   = 00000007
DISPLAY                         = 00000020                  GETSYI$_NODENAME                = 0000000C
ECASCTR                         = 0000000C                  GETTIM                          = 00000002
ENQ$_ACMODE                     = 00000028                  IDTOASC                         = 0000000A
ENQ$_ASTADR                     = 0000001C                  IMGACT                          = 00000003
ENQ$_ASTPRM                     = 00000020                  MODIFY                          = 00000024
ENQ$_BLKAST                     = 00000024                  NUMTIM                          = 00000004
ENQ$_EFN                        = 00000004                  NXTVOL                          = 00000025
ENQ$_FLAGS                      = 00000010                  OPEN                            = 00000026
ENQ$_LKMODE                     = 00000008                  PARSE                           = 00000028
ENQ$_LKSB                       = 0000000C                  PUT                             = 00000016
ENQ$_NARGS                      = 0000000B                  QIO$_ASTADR                     = 00000014
ENQ$_PARID                      = 00000018                  QIO$_ASTPRM                     = 00000018
ENQ$_PROT                       = 0000002C                  QIO$_CHAN                       = 00000008
ENQ$_RESNAM                     = 00000014                  QIO$_EFN                        = 00000004
ENTER                           = 0000002A                  QIO$_FUNC                       = 0000000C
ERASE                           = 00000021                  QIO$_IOSB                       = 00000010
EXC_MASK                        = 00000080                  QIO$_NARGS                      = 0000000C
EXTEND                          = 00000022                  QIO$_P1                         = 0000001C
FILESCAN                        = 00000034                  QIO$_P2                         = 00000020
FIND                            = 00000013                  QIO$_P3                         = 00000024
FINISH_RDB                      = 00000009                  QIO$_P4                         = 00000028
FLUSH                           = 00000023                  QIO$_P5                         = 0000002C
FREE                            = 00000014                  QIO$_P6                         = 00000030
GET                             = 00000015                  RCASCTR                         = 00000035
GETDVI$_ASTADR                  = 00000018                  RCASE                             00000004 R       02
GETDVI$_ASTPRM                  = 0000001C                  RCASMAX                         = 00000022
GETDVI$_CHAN                    = 00000008                  RCASMIN                         = 00000012
GETDVI$_DEVNAM                  = 0000000C                  READ                            = 00000017
GETDVI$_EFN                     = 00000004                  RELEASE                         = 00000018
GETDVI$_IOSB                    = 00000014                  REMOVE                          = 0000002C
GETDVI$_ITMLST                  = 00000010                  RENAME                          = 0000002D
GETDVI$_NARGS                   = 00000008                  REWIND                          = 00000027
GETDVI$_NULLARG                 = 00000020                  RMS$CLOSE                         ********    X    02
GETJPI$_ASTADR                  = 00000018                  RMS$CONNECT                       ********    X    02
GETJPI$_ASTPRM                  = 0000001C                  RMS$CREATE                        ********    X    02
GETJPI$_EFN                     = 00000004                  RMS$DELETE                        ********    X    02
GETJPI$_IOSB                    = 00000014                  RMS$DISCONNECT                    ********    X    02
GETJPI$_ITMLST                  = 00000010                  RMS$DISPATCH                      00000000 R       02
GETJP!$_NARGS                   = 00000007                  RMS$DISPLAY                       ********    X    02
GETJPI$_PIDADR                  = 00000008                  RMS$ENTER                         ********    X    02
GETJPI$_PRCNAM                  = 0000000C                  RMS$ERASE                         ********    X    02
GETLKI$_ASTADR                  = 00000014                  RMS$EXTEND                        ********    X    02
GETLKI$_ASTPRM                  = 00000018                  RMS$FILESCAN                      ********    X    02
GETLKI$_EFN                     = 00000004                  RMS$FIND                          ********    X    02
```

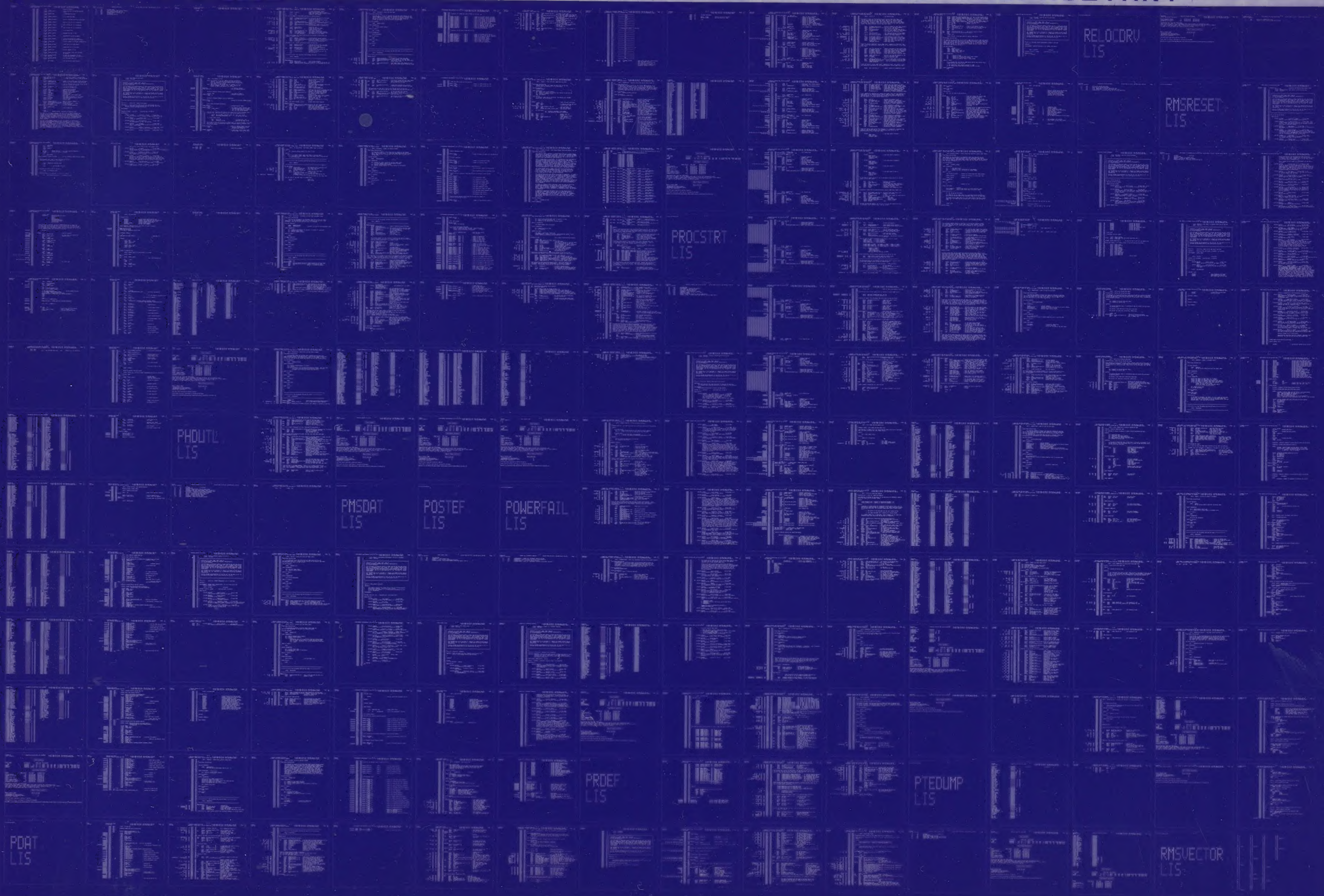
RSE
V04

```
RMS$FLUSH                        ********   X   02           UPDSEC$_RETADR                 = 00000008
RMS$FREE                         ********   X   02           UPDSEC$_UPDFLG                 = 00000010
RMS$GET                          ********   X   02           WAIT                           = 0000001A
RMS$MODIFY                       ********   X   02           WRITE                          = 0000001B
RMS$NXTVOL                       ********   X   02
RMS$OPEN                         ********   X   02
RMS$PARSE                        ********   X   02
RMS$PUT                          ********   X   02
RMS$READ                         ********   X   02
RMS$RELEASE                      ********   X   02
RMS$REMOVE                       ********   X   02
RMS$RENAME                       ********   X   02
RMS$REWIND                       ********   X   02
RMS$RMSRUHNDLR                   ********   X   02
RMS$RMSRUNDWN                    ********   X   02
RMS$SEARCH                       ********   X   02
RMS$SETDDIR                      ********   X   02
RMS$SETDFPROT                    ********   X   02
RMS$SPACE                        ********   X   02
RMS$SSVEXC                     = 00000049 R     02
RMS$TRUNCATE                     ********   X   02
RMS$UPDATE                       ********   X   02
RMS$WAIT                         ********   X   02
RMS$WRITE                        ********   X   02
RMSRUHNDLR                     = 00000033
RMSRUNDWN                      = 00000032
RMSSWITCH                      = 00000001
RUF$KCASCTR                    = 00004010
SEARCH                         = 0000002E
SETDDIR                        = 0000002F
SETDFPROT                      = 00000030
SNDACC                         = 00000007
SNDJBC                         = 00000001
SNDJBC$_ASTADR                 = 00000018
SNDJBC$_ASTPRM                 = 0000001C
SNDJBC$_EFN                    = 00000004
SNDJBC$_FUNC                   = 00000008
SNDJBC$_IOSB                   = 00000014
SNDJBC$_ITMLST                 = 00000010
SNDJBC$_NARGS                  = 00000007
SNDJBC$_NULLARG                = 0000000C
SNDOPR                         = 00000005
SNDSMB                         = 00000006
SPACE                          = 00000028
SSVEXC                         = 00000031
SYNCH$_EFN                     = 00000004
SYNCH$_IOSB                    = 00000008
SYNCH$_NARGS                   = 00000002
TRUNCATE                       = 00000029
UPDATE                         = 00000019
UPDSEC$_ACMODE                 = 0000000C
UPDSEC$_ASTADR                 = 0000001C
UPDSEC$_ASTPRM                 = 00000020
UPDSEC$_EFN                    = 00000014
UPDSEC$_INADR                  = 00000004
UPDSEC$_IOSB                   = 00000018
UPDSEC$_NARGS                  = 00000008
```

```
                                         +----------------+
                                         ! Psect synopsis !
                                         +----------------+

PSECT name                          Allocation           PSECT No.  Attributes
----------                          ----------           ---------  ----------
.  ABS  .                           00000000  (     0.)  00 (  0.)  NOPIC    USR    CON    ABS    LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                               00000000  (     0.)  01 (  1.)  NOPIC    USR    CON    ABS    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$$$RMSVEC                           0000004F  (    79.)  02 (  2.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC BYTE

                                       +------------------------+
                                       ! Performance indicators !
                                       +------------------------+

Phase                       Page faults    CPU Time       Elapsed Time
-----                       -----------    --------       ------------
Initialization                       30    00:00:00.08    00:00:00.94
Command processing                  112    00:00:00.65    00:00:06.08
Pass 1                              721    00:00:22.46    00:01:07.98
Symbol table sort                     0    00:00:01.88    00:00:03.70
Pass 2                              192    00:00:06.49    00:00:20.99
Symbol table output                  21    00:00:00.17    00:00:00.61
Psect synopsis output                 1    00:00:00.02    00:00:00.03
Cross-reference output                0    00:00:00.00    00:00:00.00
Assembler run totals               1079    00:00:31.76    00:01:40.35

The working set limit was 2100 pages.
252496 bytes (494 pages) of virtual memory were used to buffer the intermediate code.
There were 70 pages of symbol table space allocated to hold 1232 non-local and 0 local symbols.
2346 source lines were read in Pass 1, producing 15 object records in Pass 2.
44 pages of virtual memory were used to define 40 macros.

                                      +--------------------------+
                                      ! Macro library statistics !
                                      +--------------------------+

Macro library name                                Macros defined
------------------                                --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                                 6
_$255$DUA28:[SYSLIB]STARLET.MLB;2                             19
TOTALS (all libraries)                                        25

1210 GETS were required to define 25 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:RMSVECTOR/OBJ=OBJ$:RMSVECTOR MSRC$:RMSW/UPDATE=(ENH$:RMSW)+MSRC$:CMODSSDSP/UPDATE=(ENH$:CMODSSDSP)+EXECML$/LIB
```

0379 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



RELOCDRV
LIS

RMSRESET
LIS

PROCSTRT
LIS

PHDUTL
LIS

PMSDAT
LIS

POSTEF
LIS

POWERFAIL
LIS

PRDEF
LIS

PTEDUMP
LIS

PDAT
LIS

RMSVECTOR
LIS

0380 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY